

# Epidemiology of lifestyle behaviours and training injuries in British Army infantry recruits

Robinson, Mark

*Award date:*
2015

*Awarding institution:*
University of Bath


# Epidemiology of lifestyle behaviours and training injuries in British Army infantry recruits

Mark Robinson

*A thesis submitted for the degree of Doctor of Philosophy*

# Supplementary Electronic Appendix B: Derived Variable Specification

# Overview

This document provides the method (SPSS syntax) for calculating the key derived variables analysed to produce the results presented in the thesis. Alongside the Data Dictionary, it contains important information that should always be documented as part of the data management process. To aid identification of variables, the document has been split into key topics:

1. Administrative and sociodemographic characteristics
2. Physical activity
3. Diet
4. Alcohol consumption
5. Smoking
6. Co-occurrence and clustering
7. Training injuries

# Notes

- For some analyses missing values had to be converted between system missing and a defined value (-9) to enable specific analyses and variable derivation.

- Variables related to training injuries were often calculated for multiple injuries (e.g. Inj1_TrainR to Inj9_TainR). For brevity, metadata for the first variable only is provided.

# 1. Administrative and sociodemographic characteristics

## Agetrain

```
compute agetrain = ctime.days(InputDate-DoB)/365.25.
compute agetrain2 = ctime.days(StartDate_PFT-DoB)/365.25.
variable labels agetrain 'Age of recruit, calculated as number of days between d.o.b and input date'.
variable labels agetrain2 'Age of recruit, calculated as number of days between d.o.b and PFT start date'.
exe.
*NOTE: Use agetrain for analysis*.
```

## EthnicR

```
Recode Ethnic
(1=1)(2=2)(3=2)(4=2)(5=2)(6=2)(7=2)(8=2)(9=2)(10=2)(11=2)(12=2)(13=2)(14=2)
(sysmis=-9) into EthnicR.
Missing values EthnicR (-9) .
Value labels EthnicR 1 'White' 2 'Other'.
Variable labels EthnicR 'Ethnicity recoded'.
Execute.
```

## NationalityR

```
Recode Nationality (1 thru 8 = 1)(9 thru 34 = 2)(sysmis=-9) into NationalityR.
Missing values NationalityR (-9) .
Value labels NationalityR 1'British' 2 'Other'.
Variable labels NationalityR 'Nationality recoded'.
Execute.
```

## AgeGroup

```
Compute AgeGroup = -9.
If range(agetrain,16,18) AgeGroup=1.
If range(agetrain,18,20) AgeGroup=2.
If range(agetrain,20,22) AgeGroup=3.
If agetrain >=22 AgeGroup=4.
Missing values AgeGroup (-9) .
Value labels AgeGroup 1'16-17' 2'18-19' 3'20-21' 4'22 and over'.
Variable labels AgeGroup 'Age of recruit in groups'.
Execute.
```

## LossCategoryR

```
Do If SYSMIS(DischargeDate) AND SYSMIS(CompleteDate).
Compute LossCategoryR=-9.
ELSE IF (SYSMIS(DischargeDate)) AND NOT(SYSMIS(CompleteDate)) .
Compute LossCategoryR=1.
ELSE IF (SYSMIS(CompleteDate)) AND NOT(SYSMIS(DischargeDate)) .
Compute LossCategoryR=2.
End if.
Missing values LossCategoryR (-9).
Value labels LossCategoryR 1'Not discharged' 2'Discharged'.
Variable labels LossCategoryR 'Whether recruit was discharged or not'.
Execute.
```

## IncomeCat

```
*First need to calculate income scores for all 'Geocodes' in the UK.
*Statistics
Income
Valid     41772
Percentiles
20        .060
```

```
40        .090
60        .150
80        .250
*The above was drawn from a separate dataset that linked Postcodes to Geocodes to Income.

Compute IncomeCat=-9.
If range(Income,0,0.060) IncomeCat=5.
If  range(Income,0.060,0.090) IncomeCat=4.
If  range(Income,0.090,0.150) IncomeCat=3.
If  range(Income,0.150,0.250) IncomeCat=2.
If  Income>0.250 IncomeCat=1.
Value labels IncomeCat 1'1 Most deprived' 2'2' 3'3' 4'4' 5'5 Least deprived'.
Variable labels IncomeCat 'Income Quintile'.
Missing values IncomeCat (-9) .
Execute.
```

## Firstime

```
Do if LossCategoryR=1 and range(days,180,200).
Compute FirstTime=1.
Else if LossCategoryR=1 and (days<180 or days>200).
Compute FirstTime=0.
Else if LossCategoryR=2.
Compute FirstTime=0.
Else if LossCategoryR<0 or days<0.
Compute FirstTime = -9.
End if.
Variable labels FirstTime ‘Whether recruit passed out first time or not’.
Value labels FirstTime 0’No’ 1’Yes’.
Missing values FirstTime (-9).
Execute.
```

## QuestComplete

```
*First convert all system missing variables to -9* (string variables are excluded).
Recode Q1_1 Q1_2a Q1_2b Q1_2c Q1_3 Q1_4a Q1_4b Q1_4c Q1_4d Q1_4e Q1_4f  Q1_5
   Q1_6 Q2_1a  Q2_2a  Q2_3a  Q2_4a  Q3_1_1 Q3_1_2 Q3_1_3 Q3_1_4 Q3_1_5 Q3_1_6
   Q3_1_7 Q3_1_8 Q3_1_9 Q3_1_10 Q3_1_11 Q3_1_12 Q3_1_13 Q3_1_14 Q3_1_15
Q3_1_16 Q3_1_17 Q3_1_18
   Q3_1_19 Q3_1_20 Q3_1_21 Q3_1_22 Q3_1_23 Q3_1_24 Q3_2 Q3_2_1a  Q3_2_2a
Q3_2_3a
    Q3_2_4a  Q3_2_5 Q3_2_6 Q3_2_7a  Q4_1 Q4_2 Q4_3 Q5_1 Q5_2 Q5_3 Q5_4 Q5_5
Q5_6
   Q5_7 Q5_8 Q5_9 Q5_10 Q5_11 Q5_12 Q5_13 Q5_14 Q5_15 Q5_16 (sysmis=-9).
EXECUTE .

DO IF (Q1_1=-9 and Q1_2a=-9 and Q1_2b=-9 and Q1_2c=-9 and Q1_3=-9 and Q1_4a=-9 and
Q1_4b=-9 and Q1_4c=-9 and
Q1_4d=-9 and Q1_4e=-9 and Q1_4f=-9 and Q1_5=-9 and Q1_6=-9 and Q2_1a=-9 and
Q2_2a=-9  and Q2_3a=-9 and
Q2_4a=-9 and Q3_1_1=-9 and Q3_1_2=-9 and Q3_1_3=-9 and Q3_1_4=-9 and Q3_1_5=-9
and Q3_1_6=-9 and
Q3_1_7=-9 and Q3_1_8=-9 and Q3_1_9=-9 and Q3_1_10=-9 and Q3_1_11=-9 and Q3_1_12=-
9 and Q3_1_13=-9 and
Q3_1_14=-9 and Q3_1_15=-9 and Q3_1_16=-9 and Q3_1_17=-9 and Q3_1_18=-9 and
Q3_1_19=-9 and Q3_1_20=-9 and
Q3_1_21=-9 and Q3_1_22=-9 and Q3_1_23=-9 and Q3_1_24=-9 and Q3_2=-9 and Q3_2_1a=-
9 and Q3_2_2a=-9 and
Q3_2_3a=-9 and Q3_2_4a=-9 and Q3_2_5=-9 and Q3_2_6=-9 and Q3_2_7a=-9 and Q4_1=-9
and Q4_2=-9 and
Q4_3=-9 and Q5_1=-9 and Q5_2=-9 and Q5_3=-9 and Q5_4=-9 and Q5_5=-9 and Q5_6=-9
and Q5_7=-9 and Q5_8=-9
```

```
and Q5_9=-9 and Q5_10=-9 and Q5_11=-9 and Q5_12=-9 and Q5_13=-9 and Q5_14=-9 and
Q5_15=-9 and Q5_16=-9).
Compute QuestComplete = 2.
ELSE IF (Q1_1>=0 OR Q1_2a>=0 OR Q1_2b>=0 OR Q1_2c>=0 OR Q1_3>=0 OR Q1_4a>=0
OR Q1_4b>=0 OR Q1_4c>=0 OR
Q1_4d>=0 OR Q1_4e>=0 OR Q1_4f>=0 OR Q1_5>=0 OR Q1_6>=0 OR Q2_1a>=0  OR
Q2_2a>=0 OR Q2_3a>=0
OR Q2_4a>=0 OR Q3_1_1>=0 OR Q3_1_2>=0 OR Q3_1_3>=0 OR Q3_1_4>=0 OR
Q3_1_5>=0 OR Q3_1_6>=0
OR Q3_1_7>=0 OR Q3_1_8>=0 OR Q3_1_9>=0 OR Q3_1_10>=0 OR Q3_1_11>=0 OR
Q3_1_12>=0 OR Q3_1_13>=0
OR Q3_1_14>=0 OR Q3_1_15>=0 OR Q3_1_16>=0 OR Q3_1_17>=0 OR Q3_1_18>=0 OR
Q3_1_19>=0 OR Q3_1_20>=0
OR Q3_1_21>=0 OR Q3_1_22>=0 OR Q3_1_23>=0 OR Q3_1_24>=0 OR Q3_2>=0 OR
Q3_2_1a>=0  OR Q3_2_2a>=0
OR Q3_2_3a>=0 OR Q3_2_4a>=0 OR Q3_2_5>=0 OR Q3_2_6>=0 OR Q3_2_7a>=0 OR
Q4_1>=0 OR Q4_2>=0
OR Q4_3>=0 OR Q5_1>=0 OR Q5_2>=0 OR Q5_3>=0 OR Q5_4>=0 OR Q5_5>=0 OR
Q5_6>=0 OR Q5_7>=0
OR Q5_8>=0 OR Q5_9>=0 OR Q5_10>=0 OR Q5_11>=0 OR Q5_12>=0 OR Q5_13>=0 OR
Q5_14>=0 OR Q5_15>=0
OR Q5_16>=0).
Compute QuestComplete = 1.
END IF.
Value labels QuestComplete 1'Yes' 2'No'.
EXECUTE .

*Compare new variables with those from previous dataset*
*all output matches*.
```

## ExitComplete

```
Do If SYSMIS(Exit1) AND SYSMIS(Exit2) AND SYSMIS(Exit3) AND SYSMIS(Exit4) AND
SYSMIS(Exit5) AND SYSMIS(Exit6)  AND SYSMIS(Exit7) AND SYSMIS(Exit8) and
SYSMIS(Exit9) AND
SYSMIS(Exit10) AND SYSMIS(Exit11) AND SYSMIS(Exit12) AND SYSMIS(Exit13) AND
SYSMIS(Exit14)  AND
SYSMIS(Exit15) AND SYSMIS(Exit16) .
Compute ExitComplete=2.
ELSE IF Exit1>0 or Exit2>0 or Exit3>0 or Exit4>0 or Exit5>0 or Exit6>0 or Exit7>0 or Exit8>0 or
Exit9>0 or Exit10>0 or Exit11>0 or Exit12>0 or Exit13>0 or Exit14>0 or Exit15>0 or Exit16>0.
Compute ExitComplete=1.
END IF.
Value labels ExitComplete 1'Yes' 2'No'.
Variable labels ExitComplete 'Whether exit questionnaire was completed or not'.
EXECUTE.
```

## Season

```
Compute StartMonth = xdate.month(StartDate_PFT).
Execute.

Do if StartMonth=12 or StartMonth=1 or StartMonth=2.
Compute Season=1.
Else if StartMonth=3 or StartMonth=4 or StartMonth=5.
Compute Season=2.
Else if StartMonth=6 or StartMonth=7 or StartMonth=8.
Compute Season=3.
Else if StartMonth=9 or StartMonth=10 or StartMonth=11.
Compute Season=4.
Else.
End if.
```

```
Value labels Season 1'Winter' 2'Spring' 3'Summer' 4'Winter'.
Variable labels Season 'Season recruit started training'.
```

# 2. Physical activity

## LTEQ

```
*Syntax to recode system missing variables to 0 when there are values in the other LTEQ
response options. If all 3 categories are blank then these remain as system missing*.
DO IF SYSMIS(Q1_2a) AND (NOT(SYSMIS(Q1_2b)) or NOT(SYSMIS(Q1_2c) ) ).
RECODE Q1_2a (SYSMIS=0)  .
ELSE IF SYSMIS(Q1_2b) AND (NOT(SYSMIS(Q1_2a)) or NOT(SYSMIS(Q1_2c) ) ).
RECODE Q1_2b (SYSMIS=0)  .
ELSE IF SYSMIS(Q1_2c) AND (NOT(SYSMIS(Q1_2a)) or NOT(SYSMIS(Q1_2b) ) ).
RECODE Q1_2c (SYSMIS=0)  .
END IF.
EXECUTE .

*Syntax to compute LTEQ score*.
compute LTEQ=(Q1_2a*9)+(Q1_2b*5)+(Q1_2c*3).
variable labels LTEQ 'Total LTEQ Score'.
Execute.
```

## Q1_1R

```
Recode Q1_1 (1=1)(2=1)(3=2)(4=3)(5=3) into Q1_1R.
Value labels Q1_1R 1'Less active' 2'Same' 3'More active'.
Missing values Q1_1R (-9).
Execute.
```

## misclassQ1_1

```
Do if (RunTot1_Q=1 OR RunTot1_Q=2) and (Q1_1=1 or Q1_1=2).
Compute misclassQ1_1=1.
Else if (RunTot1_Q=1 OR RunTot1_Q=2) and range(Q1_1,3,5).
Compute misclassQ1_1=0.
Else if range (RunTot1_Q,3,5).
Compute misclassQ1_1=0.
End If.
Variable labels misclassQ1_1 'Misclassification between run time quintiles and Q1_1 category'.
Value labels 0'Not misclassified' 1'Misclassified'.
```

## misclassLTEQ

```
Do if (RunTot1_Q=1 OR RunTot1_Q=2) and (LTEQ_Q=4 or LTEQ_Q=5).
Compute misclassLTEQ=1.
Else if (RunTot1_Q=1 OR RunTot1_Q=2) and range(LTEQ_Q,1,3).
Compute misclassLTEQ =0.
Else if range (RunTot1_Q,3,5).
Compute misclassLTEQ =0.
End If.
Variable labels misclassQ1_1 'Misclassification between run time quintiles and LTEQ quintile'.
Value labels 0'Not misclassified' 1'Misclassified'.
```

# 3. Diet

## REAP

```
*Create total REAP score - assigning the value of 3 to missing values*.
```

```
Do if  (q3_1_1R>0 or Q3_1_2R>0 or Q3_1_3R>0 or Q3_1_4R>0 or Q3_1_5R>0 or Q3_1_6R>0
or Q3_1_7R>0
 or Q3_1_8R>0 or Q3_1_9R>0 or Q3_1_10R>0 or Q3_1_11R>0 or Q3_1_12R>0 or
Q3_1_13R>0 or Q3_1_14R>0
 or Q3_1_15R>0 or Q3_1_16R>0 or Q3_1_17R>0 or Q3_1_18R>0  or Q3_1_19R>0 or
Q3_1_20R>0
or Q3_1_21R>0 or Q3_1_22R>0 or Q3_1_23R>0 or Q3_1_24R>0).
Recode Q3_1_1R Q3_1_2R Q3_1_3R Q3_1_4R Q3_1_5R Q3_1_6R Q3_1_7R Q3_1_8R
Q3_1_9R  Q3_1_10R
Q3_1_11R Q3_1_12R  Q3_1_13R  Q3_1_14R  Q3_1_15R  Q3_1_16R Q3_1_17R Q3_1_18R
Q3_1_19R
Q3_1_20R Q3_1_21R Q3_1_22R  Q3_1_23R Q3_1_24R (-9=3).
Else if  (q3_1_1R=-9 and Q3_1_2R=-9 and Q3_1_3R=-9 and Q3_1_4R=-9 and Q3_1_5R=-9
and Q3_1_6R=-9 and Q3_1_7R=-9
 and Q3_1_8R=-9 and Q3_1_9R=-9 and Q3_1_10R=-9 and Q3_1_11R=-9 and Q3_1_12R=-9
and Q3_1_13R=-9 and Q3_1_14R=-9
 and Q3_1_15R=-9 and Q3_1_16R=-9 and Q3_1_17R=-9 and Q3_1_18R=-9  and Q3_1_19R=-
9 and Q3_1_20R=-9
and Q3_1_21R=-9 and Q3_1_22R=-9 and Q3_1_23R=-9 and Q3_1_24R=-9).
Compute REAP_R=-9.
END IF.

Compute REAP_R = Q3_1_1R + Q3_1_2R + Q3_1_3R + Q3_1_4R + Q3_1_5R
 + Q3_1_6R + Q3_1_7R + Q3_1_8R + Q3_1_9R + Q3_1_10R + Q3_1_11R
+ Q3_1_12R + Q3_1_13R + Q3_1_14R + Q3_1_15R + Q3_1_16R + Q3_1_17R
 + Q3_1_18R + Q3_1_19R + Q3_1_20R + Q3_1_21R + Q3_1_22R + Q3_1_23R
 + Q3_1_24R.
Execute.
```

# 4. Alcohol consumption

### AUDIT_C

```
*Recode original variables so that values are consistent with scoring system*.
Recode Q4_1 (1=0) (2=1)(3=2)(4=3)(5=4) into Q4_1Recode.
Recode Q4_2 (1=0)(2=0)(3=1)(4=2)(5=3)(6=4) into Q4_2Recode.
Recode Q4_3 (1=0)(2=1)(3=2)(4=3)(5=4) into Q4_3Recode.
variable labels
Q4_1Recode 'Frequency of alcohol consumption in past year (recoded for calculation of AUDIT
score )'
Q4_2Recode 'Number of drinks on typical day/night (recoded for calculation of AUDIT score)'
Q4_3Recode 'Frequency of consumption of 6 or more drinks (recoded for calculation of AUDIT
score)'.
execute.

*Create variable that calculates total AUDIT score*.
compute AUDIT_C = Q4_1Recode + Q4_2Recode + Q4_3Recode.
variable labels AUDIT_C 'Total AUDIT-C Score'.
Execute.

*Check frequencies of original and new variables*.
FREQUENCIES VARIABLES= Q4_1 Q4_1Recode Q4_2 Q4_2Recode Q4_3 Q4_3Recode
AUDIT_C
  /ORDER=ANALYSIS.
Execute.
```

# 5. Smoking

### CDS_5

```
*Recode first 3 CDS-5 question responses: variables Q5_6 Q5_7 Q5_8.
Compute Q5_6R=Q5_6.
If range(Q5_6,0,20) Q5_6R=1.
If range(Q5_6,21,40) Q5_6R=2.
If range(Q5_6,41,60) Q5_6R=3.
If range(Q5_6,61,80) Q5_6R=4.
If range(Q5_6,81,100) Q5_6R=5.
VARIABLE LABELS Q5_6R 'CDS1 Recoded'.
Execute.

Compute Q5_7R=Q5_7.
If range(Q5_7,0,5) Q5_7R=1.
If range(Q5_7,6,10) Q5_7R=2.
If range(Q5_7,11,20) Q5_7R=3.
If range(Q5_7,21,29) Q5_7R=4.
If Q5_7>29 Q5_7R=5.
VARIABLE LABELS Q5_7R 'CDS2 Recoded'.
Execute.


Recode Q5_6R (sysmis=-9).
Recode Q5_7R (sysmis=-9).
Recode Q5_8 (1=5)(2=4)(3=3)(4=2)(5=1)(sysmis=-9) into Q5_8R.
Recode Q5_9 (1=1)(2=2)(3=3)(4=4)(5=5)(sysmis=-9) into Q5_9R.
Recode Q5_10 (1=1)(2=2)(3=3)(4=4)(5=5)(sysmis=-9) into Q5_10R.
VARIABLE LABELS Q5_9R 'CDS4 Recoded' Q5_10R 'CDS5 Recoded'.
Execute.

Recode Q5_8R (sysmis=-9).
VARIABLE LABELS Q5_8R 'CDS3 Recoded'.
Execute.

*Create total CDS-5 score*.
Do if (Q5_6R=-9 or Q5_7R=-9 or Q5_8R=-9 or Q5_9R=-9 or Q5_10R=-9).
Compute CDS_5=-9.
Else if (Q5_6R>0 and Q5_7R>0 and Q5_8R>0 and Q5_9R>0 and Q5_10R>0).
Compute CDS_5 = Q5_6R + Q5_7R + Q5_8R + Q5_9R + Q5_10R.
End IF.
VARIABLE LABELS CDS_5 'Total score on Cigareete Dependency Scale'.
Execute.
```

## CDS_5Ex

```
*Recode first 3 CDS-5 question responses: variables Exit6 Exit7 Exit8.
Compute Exit6R=Exit6.
If range(Exit6,0,20) Exit6R=1.
If range(Exit6,21,40) Exit6R=2.
If range(Exit6,41,60) Exit6R=3.
If range(Exit6,61,80) Exit6R=4.
If range(Exit6,81,100) Exit6R=5.
VARIABLE LABELS Exit6R 'CDS1 Exit Recoded'.
Execute.

Compute Exit7R=Exit7.
If range(Exit7,0,5) Exit7R=1.
If range(Exit7,6,10) Exit7R=2.
If range(Exit7,11,20) Exit7R=3.
If range(Exit7,21,29) Exit7R=4.
If Exit7>29 Exit7R=5.
VARIABLE LABELS Q5_7R 'CDS2 Recoded'.
Execute.
```

```
Recode Exit6R (sysmis=-9).
Recode Exit7R (sysmis=-9).
Recode Exit8 (1=5)(2=4)(3=3)(4=2)(5=1)(sysmis=-9) into Exit8R.
Recode Exit9 (1=1)(2=2)(3=3)(4=4)(5=5)(sysmis=-9) into Exit9R.
Recode Exit10 (1=1)(2=2)(3=3)(4=4)(5=5)(sysmis=-9) into Exit10R.
VARIABLE LABELS Q5_9R 'CDS4 exit Recoded' Q5_10R 'CDS5 exit Recoded'.
Execute.

Recode Exit8R (sysmis=-9).
VARIABLE LABELS Exit8R 'CDS3 exit Recoded'.
Execute.

*Create total CDS-5 score*.
Do if (Exit6R=-9 or Exit7R=-9 or Exit8R=-9 or Exit9R=-9 or Exit10R=-9).
Compute CDS_5=-9.
Else if (Exit6R>0 and Exit7R>0 and Exit8R>0 and Exit9R>0 and Exit10R>0).
Compute CDS_5Ex = Exit6R + Exit7R + Exit8R + Exit9R + Exit10R.
End IF.
VARIABLE LABELS CDS_5Ex 'Total score on Cigarette Dependency Scale'.
Execute.
```

## Q5_2Diff, Q5_6Diff and Q5_7Diff

```
Do if (Q5_2<0) or (Exit2<0).
Compute Q5_2Diff=-9.
Else if (Q5_2>=0) OR (Exit2>=0).
Compute Q5_2Diff=Q5_2-Exit2.
End if.
Execute.
Variable labels Q5_2Diff 'Q5_2Difference between MPQ and Exit Questionnaire'.

Do if (Q5_6<0) or (Exit6<0).
Compute Q5_6Diff=-9.
Else if (Q5_6>=0) OR (Exit6>=0).
Compute Q5_6Diff=Q5_6-Exit6.
End if.
Execute.
Variable labels Q5_6Diff 'Q5_6Difference between MPQ and Exit Questionnaire'.

Do if (Q5_7<0) or (Exit7<0).
Compute Q5_7Diff=-9.
Else if (Q5_7>=0) OR (Exit7>=0).
Compute Q5_7Diff=Q5_7-Exit7.
End if.
Execute.
Variable labels Q5_7Diff 'Q5_7 Difference between MPQ and Exit Questionnaire'.

Do if (CDS_5<0) or (CDS_5Ex<0).
Compute CDS_5Diff=-9.
Else if (CDS_5>=0) OR (CDS_5Ex>=0).
Compute CDS_5Diff=CDS_5-CDS_5Ex.
End if.
Execute.
Variable labels CDS_5Diff 'CDS_5Difference between MPQ and Exit Questionnaire'.
```

## Smoking

```
Do if (Q5_2<0) or (Exit2<0).
o If Q5_1=1 and Q5_3=1 and Q5_4=1.
Compute Smoking=1.
Else if Q5_1=1 and Q5_3=1 and Q5_4=2.
Compute Smoking=2.
```

```
Else if Q5_1=2.
Compute Smoking=3.
Else if Q5_1=1 and Q5_3=2.
Compute Smoking=4.
Else if Q5_1<0 and Q5_3<0 and Q5_4<0.
Compute Smoking=-9.
End if.
Variable labels Smoking 'Smoking status'.
Value labels Smoking 1'Regular smoker' 2'Occasional smoker' 3'Never smoker' 4'Ex-smoker'.
execute.
```

## SmokingR

```
Recode Smoking (1=1)(2=1)(3=2)(4=2)(-9=-9) Into SmokingR.
Value labels SmokingR 1'Current smoker' 2'Not current smoker' .
Variable labels SmokingR 'Smoking status - current or not'.
Execute.
```

## Smoking3

```
Do if (Q5_2<0) or (Exit2<0).
Recode Smoking (1=1)(2=1)(3=3)(4=2) into Smoking3.
Value labels Smoking3 1'Current smoker' 2'Ex smoker' 3'Never smoker'.
Variable labels Smoking3 'Smoking status - 3 levels'.
Execute.
fre Smoking3.
*x-checked with Smoking and frequencies match.
```

## NumberCigs

```
Do if Smoking=3.
Compute NumberCigs=0.
Else if Smoking=4.
Compute NumberCigs=1.
Else if (Smoking=1 or smoking=2) and range(Q5_7,0,10).
Compute NumberCigs=2.
Else if (Smoking=1 or smoking=2) and Q5_7>10.
Compute NumberCigs=3.
Else if Smoking<0.
Compute NumberCigs=-9.
End if.
Value labels NumberCigs 0 'Never smoker' 1'Ex-smoker' 2'1-10' 3'>10'.
Variable labels NumberCigs 'Number of cigarettes smoked per day'.
Execute.
fre NumberCigs.
*x-checked with Smoking and frequencies match.
```

## NumberCigsR

```
Recode NumberCigs (0=1)(1=1)(2=2)(3=3) into NumberCigsR.
Value labels NumberCigsR 1'Non-smoker / ex-smoker' 2'1-10' 3'>10'.
Variable labels NumberCigsR 'Number of cigarettes smoked per day recoded'.
Execute.
fre NumbercigsR.
*x-checked with Smoking and frequencies match.
```

## PackYears

```
Do if NumberCigs=0 or NumberCigs=1.
Compute PackYears=-9.
Else if NumberCigs>1 AND q5_2>=5.
Compute PackYears=((agetrain-Q5_2)*Q5_7)/20.
Else if NumberCigs>1 AND range(q5_2,0,4).
Compute PackYears=-9.
```

```
End if.
Missing values PackYears(-9).
Execute.
```

### PackYears3

```
Do if NumberCigs=0 or NumberCigs=1.
Compute PackYears3=0.
Else if PackYears_Q3=1.
Compute PackYears3=1.
Else if PackYears_Q3=2.
Compute PackYears3=2.
Else if PackYears_Q3=3.
Compute PackYears3=3.
End if.
```

## 6. Co-occurrence and clustering

### SmokeScore

```
*Both regular and occasional smokers to be included as current smokers.
*Use SmokingR variable.
Compute SmokeScore=SmokingR.
Recode SmokeScore (2=0).
Execute.
fre SmokeScore.
*cross-checked with original smoking variable and they match.
```

### Q1_1Less

```
DO IF range(Q1_1,1,2).
COMPUTE Q1_1Less=1.
ELSE IF Q1_1>2.
COMPUTE Q1_1Less=0.
ELSE IF Q1_1<0.
COMPUTE Q1_1Less=-9.
END IF.
Variable labels Q1_1Less 'Self-reported less activity than others of same age'.
Value labels Q1_1Less 0'No' 1'Yes'.
Missing values Q1_1Less (-9) .
EXECUTE .

*Check frequencies.
FREQUENCIES VARIABLES=Q1_1Less
  /ORDER=ANALYSIS.
*Frequencies ok*.
```

### ProbDrk

```
*Reinart and Allen suggest cut of 5+ - given so many of the sample fall into this category, use
this cutoff.
Do if AUDIT_C>4.
Compute ProbDrk=1.
Else if AUDIT_C<0.
Compute ProbDrk=-9.
Else.
Compute ProbDrk=0.
End if.
```

```
Value labels ProbDrk 1'Harmful drinker (AUDIT>=5)' 0'Not problem drinker'.
Variable labels ProbDrk 'Whether recruit has AUDIT score >=5'.
Execute.
fre ProbDrk.
*cross-checked with previous analyses and it matches.
```

## DietR

```
*Use Fruit and Vegetable question - check % of those rarely eating 2-3 portions of fruit and 3-4 portions of vegetables.
Do if Q3_1_4=3 AND Q3_1_5=3.
Compute Diet=1.
Else if Q3_1_4<0 or Q3_1_5<0.
Compute Diet=-9.
Else.
Compute Diet=0.
End if.
Value labels Diet 0'Often or sometimes eats fruit and veg' 1'Rarely eats fruit and veg'.
Variable labels Diet 'Whether recruits rarely eats fruit and vegetables (from REAP)'.
Execute.

*Use Fruit and Vegetable question - check % of those rarely or sometimes eating 2-3 portions of fruit and 3-4 portions of vegetables.
Do if (Q3_1_4=3 or Q3_1_4=2) AND (Q3_1_5=2 OR Q3_1_5=3).
Compute DietR=1.
Else if Q3_1_4<0 or Q3_1_5<0.
Compute DietR=-9.
Else.
Compute DietR=0.
End if.
Value labels DietR 0'Often or sometimes eats fruit and veg' 1 'rarely eats fruit and veg'.
Variable labels DietR 'Whether recruits sometimes or rarely eats fruit and vegetables (from REAP)'.
Execute.
```

## HealthScore

```
Compute HealthScore=SmokeScore + ProbDrk + DietR + Q1_1Less.
Execute.
Value labels HealthScore 0'0 risk factors' 1'1Risk factor' 2'2Risk factors' 3'3 risk factors' 4'4 risk factors'.
Variable labels HealthScore 'Recruits health score (0-4 risk factors)'.
Execute.
fre HealthScore.
```

## SmokeOnly

```
*Smoking only.
Do if SmokeScore=1 and ProbDrk=0 and DietR=0 and Q1_1Less=0.
Compute SmokeOnly=1.
Else if SmokeScore=1 and (ProbDrk=0 or ProbDrk=1) and DietR=1 and (Q1_1Less=0 or Q1_1Less=1).
Compute SmokeOnly=0.
Else if SmokeScore=1 and ProbDrk=1 and (DietR=0 or DietR=1) and (Q1_1Less=0 or Q1_1Less=1).
Compute SmokeOnly=0.
Else if SmokeScore=1 and (ProbDrk=0 or ProbDrk=1) and (DietR=0 or DietR=1) and Q1_1Less=1.
Compute SmokeOnly=0.
Else if SmokeScore=0 and (ProbDrk=0 or ProbDrk=1) and (DietR=0 or DietR=1) and (Q1_1Less=0 or Q1_1Less=1).
Compute SmokeOnly=0.
End if.
```

```
Value labels SmokeOnly 0 'Doesnt smoke or have any other risk factor' 1'Smokes only'.
Variable labels SmokeOnly 'Recruit only smokes and has no other risk factor'.
Execute.
fre SmokeOnly.
*total n the same as total n for HealthScore.
```

## PAOnly

```
*PA only.
Do if Q1_1Less=1 and ProbDrk=0 and DietR=0 and SmokeScore=0.
Compute PAOnly=1.
Else if Q1_1Less=1 and (ProbDrk=0 or ProbDrk=1) and DietR=1 and (SmokeScore=0 or
SmokeScore=1).
Compute PAOnly=0.
Else if Q1_1Less=1 and ProbDrk=1 and (DietR=0 or DietR=1) and (SmokeScore=0 or
SmokeScore=1).
Compute PAOnly=0.
Else if Q1_1Less=1 and (ProbDrk=0 or ProbDrk=1) and (DietR=0 or DietR=1) and
SmokeScore=1.
Compute PAOnly=0.
Else if Q1_1Less=0 and (ProbDrk=0 or ProbDrk=1) and (DietR=0 or DietR=1) and
(SmokeScore=0 or SmokeScore=1).
Compute PAOnly=0.
End if.
Value labels PAOnly 0 'Active and has no other risk factor' 1'Inactive only'.
Variable labels PAOnly 'Recruit only inactive and has no other risk factor'.
Execute.
fre PAOnly.
*total n the same as total n for HealthScore.
```

## AlcOnly

```
*Alcohol only.
Do if ProbDrk=1 and Q1_1Less=0 and DietR=0 and SmokeScore=0.
Compute AlcOnly=1.
Else if ProbDrk=1 and (Q1_1Less=0 or Q1_1Less=1) and DietR=1 and (SmokeScore=0 or
SmokeScore=1).
Compute AlcOnly=0.
Else if ProbDrk=1 and Q1_1Less=1 and (DietR=0 or DietR=1) and (SmokeScore=0 or
SmokeScore=1).
Compute AlcOnly=0.
Else if ProbDrk=1 and (Q1_1Less=0 or Q1_1Less=1) and (DietR=0 or DietR=1) and
SmokeScore=1.
Compute AlcOnly=0.
Else if ProbDrk=0 and (Q1_1Less=0 or Q1_1Less=1) and (DietR=0 or DietR=1) and
(SmokeScore=0 or SmokeScore=1).
Compute AlcOnly=0.
End if.
Value labels AlcOnly 0 'Not harmful drinker and has no other risk factor' 1'Harmful drinker only'.
Variable labels AlcOnly 'Recruit only harmful drinker and has no other risk factor'.
Execute.
fre AlcOnly.
*total n the same as total n for HealthScore.
```

## DietOnly

```
*Diet only.
Do if DietR=1 and Q1_1Less=0 and ProbDrk=0 and SmokeScore=0.
Compute DietOnly=1.
Else if DietR=1 and (Q1_1Less=0 or Q1_1Less=1) and ProbDrk=1 and (SmokeScore=0 or
SmokeScore=1).
Compute DietOnly=0.
```

```
Else if DietR=1 and Q1_1Less=1 and (ProbDrk=0 or ProbDrk=1) and (SmokeScore=0 or
SmokeScore=1).
Compute DietOnly=0.
Else if DietR=1 and (Q1_1Less=0 or Q1_1Less=1) and (ProbDrk=0 or ProbDrk=1) and
SmokeScore=1.
Compute DietOnly=0.
Else if DietR=0 and (Q1_1Less=0 or Q1_1Less=1) and (ProbDrk=0 or ProbDrk=1) and
(SmokeScore=0 or SmokeScore=1).
Compute DietOnly=0.
End if.
Value labels DietOnly 0 'Eats fruit and veg and has no other risk factor' 1'Low fruit and veg only'.
Variable labels DietOnly 'Recruit only low fruit and veg and has no other risk factor'.
Execute.
fre DietOnly.
*total n the same as total n for HealthScore.
```

## SmokeDrink

```
*Smoking and Drinking.
Do if SmokeScore=1 and ProbDrk=1 and DietR=0 and Q1_1Less=0.
Compute SmokeDrink=1.
Else if SmokeScore=1 and ProbDrk=1 and (DietR=0 or DietR=1) and Q1_1Less=1.
Compute SmokeDrink=0.
Else if SmokeScore=1 and ProbDrk=1 and DietR=1 and (Q1_1Less=0 or Q1_1Less=1).
Compute SmokeDrink=0.
Else if SmokeScore=0 and (ProbDrk=0 or ProbDrk=1) and (DietR=0 or DietR=1) and
(Q1_1Less=0 or Q1_1Less=1).
Compute SmokeDrink=0.
Else if (SmokeScore=0 or SmokeScore=1) and ProbDrk=0 and (DietR=0 or DietR=1) and
(Q1_1Less=0 or Q1_1Less=1).
Compute SmokeDrink=0.
End if.
Value labels SmokeDrink 0 'No combination' 1'Smokes and harmful drinker'.
Variable labels SmokeDrink 'Recruit smokes and is harmful drinker but has no other risk factor'.
Execute.
fre SmokeDrink.
*total n the same as total n for HealthScore.
```

## SmokePA

```
*Smoking and PA.
Do if SmokeScore=1 and Q1_1Less=1 and DietR=0 and ProbDrk=0.
Compute SmokePA=1.
Else if SmokeScore=1 and Q1_1Less=1 and (DietR=0 or DietR=1) and ProbDrk=1.
Compute SmokePA=0.
Else if SmokeScore=1 and Q1_1Less=1 and DietR=1 and (ProbDrk=0 or ProbDrk=1).
Compute SmokePA=0.
Else if SmokeScore=0 and (Q1_1Less=0 or Q1_1Less=1) and (DietR=0 or DietR=1) and
(ProbDrk=0 or ProbDrk=1).
Compute SmokePA=0.
Else if (SmokeScore=0 or SmokeScore=1) and Q1_1Less=0 and (DietR=0 or DietR=1) and
(ProbDrk=0 or ProbDrk=1).
Compute SmokePA=0.
End if.
Value labels SmokePA 0 'No combination' 1'Smokes and inactive'.
Variable labels SmokePA 'Recruit smokes and is inactive but has no other risk factor'.
Execute.
fre SmokePA.
*total n the same as total n for HealthScore.
```

## SmokeDiet

```
*Smoking and Diet.
```

```
Do if SmokeScore=1 and DietR=1 and Q1_1Less=0 and ProbDrk=0.
Compute SmokeDiet=1.
Else if SmokeScore=1 and DietR=1 and (Q1_1Less=0 or Q1_1Less=1) and ProbDrk=1.
Compute SmokeDiet=0.
Else if SmokeScore=1 and DietR=1 and Q1_1Less=1 and (ProbDrk=0 or ProbDrk=1).
Compute SmokeDiet=0.
Else if SmokeScore=0 and (DietR=0 or DietR=1) and (Q1_1Less=0 or Q1_1Less=1) and
(ProbDrk=0 or ProbDrk=1).
Compute SmokeDiet=0.
Else if (SmokeScore=0 or SmokeScore=1) and DietR=0 and (Q1_1Less=0 or Q1_1Less=1) and
(ProbDrk=0 or ProbDrk=1).
Compute SmokeDiet=0.
End if.
Value labels SmokeDiet 0 'No combination' 1'Smokes and low fruit'.
Variable labels SmokeDiet 'Recruit smokes and has low fruit intake but has no other risk factor'.
Execute.
fre SmokeDiet.
*total n the same as total n for HealthScore.
```

## DrinkDiet

```
*Drinking and Diet.
Do if ProbDrk=1 and DietR=1 and Q1_1Less=0 and SmokeScore=0.
Compute DrinkDiet=1.
Else if ProbDrk=1 and DietR=1 and (Q1_1Less=0 or Q1_1Less=1) and SmokeScore=1.
Compute DrinkDiet=0.
Else if ProbDrk=1 and DietR=1 and Q1_1Less=1 and (SmokeScore=0 or SmokeScore=1).
Compute DrinkDiet=0.
Else if ProbDrk=0 and (DietR=0 or DietR=1) and (Q1_1Less=0 or Q1_1Less=1) and
(SmokeScore=0 or SmokeScore=1).
Compute DrinkDiet=0.
Else if (ProbDrk=0 or ProbDrk=1) and DietR=0 and (Q1_1Less=0 or Q1_1Less=1) and
(SmokeScore=0 or SmokeScore=1).
Compute DrinkDiet=0.
End if.
Value labels DrinkDiet 0 'No combination' 1'Drinks harmfully and low fruit'.
Variable labels DrinkDiet 'Recruit drinks harmfully and has low fruit intake but has no other risk
factor'.
Execute.
fre DrinkDiet.
*total n the same as total n for HealthScore.
```

## DrinkPA

```
*Drinking and Activity.
Do if ProbDrk=1 and Q1_1Less=1 and DietR=0 and SmokeScore=0.
Compute DrinkPA=1.
Else if ProbDrk=1 and Q1_1Less=1 and (DietR=0 or DietR=1) and SmokeScore=1.
Compute DrinkPA=0.
Else if ProbDrk=1 and Q1_1Less=1 and DietR=1 and (SmokeScore=0 or SmokeScore=1).
Compute DrinkPA=0.
Else if ProbDrk=0 and (Q1_1Less=0 or Q1_1Less=1) and (DietR=0 or DietR=1) and
(SmokeScore=0 or SmokeScore=1).
Compute DrinkPA=0.
Else if (ProbDrk=0 or ProbDrk=1) and Q1_1Less=0 and (DietR=0 or DietR=1) and
(SmokeScore=0 or SmokeScore=1).
Compute DrinkPA=0.
End if.
Value labels DrinkPA 0 'No combination' 1'Drinks harmfully and low activity'.
Variable labels DrinkPA 'Recruit drinks harmfully and has low activity but has no other risk
factor'.
```

```
Execute.
fre DrinkPA.
*total n the same as total n for HealthScore.
```

## DietPA

```
*Diet and Activity.
Do if DietR=1 and Q1_1Less=1 and ProbDrk=0 and SmokeScore=0.
Compute DietPA=1.
Else if DietR=1 and Q1_1Less=1 and (ProbDrk=0 or ProbDrk=1) and SmokeScore=1.
Compute DietPA=0.
Else if DietR=1 and Q1_1Less=1 and ProbDrk=1 and (SmokeScore=0 or SmokeScore=1).
Compute DietPA=0.
Else if DietR=0 and (Q1_1Less=0 or Q1_1Less=1) and (ProbDrk=0 or ProbDrk=1) and
(SmokeScore=0 or SmokeScore=1).
Compute DietPA=0.
Else if (DietR=0 or DietR=1) and Q1_1Less=0 and (ProbDrk=0 or ProbDrk=1) and
(SmokeScore=0 or SmokeScore=1).
Compute DietPA=0.
End if.
Value labels DietPA 0 'No combination' 1'low fruit and activity'.
Variable labels DietPA 'Recruit has low fruit/veg activity but has no other risk factor'.
Execute.
fre DietPA.
*total n the same as total n for HealthScore.
```

## SmokeDrinkDiet

```
*Smoking, drinking and Diet.
Do if SmokeScore=1 and ProbDrk=1 and DietR=1 and Q1_1Less=0.
Compute SmokeDrinkDiet=1.
Else if SmokeScore=1 and ProbDrk=1 and DietR=1 and Q1_1Less=1.
Compute SmokeDrinkDiet=0.
Else if SmokeScore=0 and (ProbDrk=0 or ProbDrk=1) and (DietR=0 or DietR=1) and
(Q1_1Less=0 or Q1_1Less=1).
Compute SmokeDrinkDiet=0.
Else if (SmokeScore=0 or SmokeScore=1) and ProbDrk=0 and (DietR=0 or DietR=1) and
(Q1_1Less=0 or Q1_1Less=1).
Compute SmokeDrinkDiet=0.
Else if (SmokeScore=0 or SmokeScore=1) and (ProbDrk=0 or ProbDrk=1) and DietR=0 and
(Q1_1Less=0 or Q1_1Less=1).
Compute SmokeDrinkDiet=0.
End if.
Value labels SmokeDrinkDiet 0 'No combination' 1'Smokes,harmful drinker and low fruit'.
Variable labels SmokeDrinkDiet 'Recruit smokes, is harmful drinker and low fruit but is active'.
Execute.
fre SmokeDrinkDiet.
*total n the same as total n for HealthScore.
```

## SmokeDrinkPA

```
*Smoking, drinking and PA.
Do if SmokeScore=1 and ProbDrk=1 and Q1_1Less=1 and DietR=0.
Compute SmokeDrinkPA=1.
Else if SmokeScore=1 and ProbDrk=1 and Q1_1Less=1 and DietR=1.
Compute SmokeDrinkPA=0.
Else if SmokeScore=0 and (ProbDrk=0 or ProbDrk=1) and (Q1_1Less=0 or Q1_1Less=1) and
(DietR=0 or DietR=1).
Compute SmokeDrinkPA=0.
Else if (SmokeScore=0 or SmokeScore=1) and ProbDrk=0 and (Q1_1Less=0 or Q1_1Less=1)
and (DietR=0 or DietR=1).
Compute SmokeDrinkPA=0.
```

```
Else if (SmokeScore=0 or SmokeScore=1) and (ProbDrk=0 or ProbDrk=1) and Q1_1Less=0
and (DietR=0 or DietR=1).
Compute SmokeDrinkPA=0.
End if.
Value labels SmokeDrinkPA 0 'No combination' 1'Smokes,harmful drinker and inactive'.
Variable labels SmokeDrinkPA 'Recruit smokes, is harmful drinker and inactive but eats fruit'.
Execute.
fre SmokeDrinkPA.
*total n the same as total n for HealthScore.
```

## DietDrinkPA

```
*Diet, drinking and PA.
Do if DietR=1 and ProbDrk=1 and Q1_1Less=1 and SmokeScore=0.
Compute DietDrinkPA=1.
Else if DietR=1 and ProbDrk=1 and Q1_1Less=1 and SmokeScore=1.
Compute DietDrinkPA=0.
Else if DietR=0 and (ProbDrk=0 or ProbDrk=1) and (Q1_1Less=0 or Q1_1Less=1) and
(SmokeScore=0 or SmokeScore=1).
Compute DietDrinkPA=0.
Else if (DietR=0 or DietR=1) and ProbDrk=0 and (Q1_1Less=0 or Q1_1Less=1) and
(SmokeScore=0 or SmokeScore=1).
Compute DietDrinkPA=0.
Else if (DietR=0 or DietR=1) and (ProbDrk=0 or ProbDrk=1) and Q1_1Less=0 and
(SmokeScore=0 or SmokeScore=1).
Compute DietDrinkPA=0.
End if.
Value labels DietDrinkPA 0 'No combination' 1'Low fruit,harmful drinker and inactive'.
Variable labels DietDrinkPA 'Low fruit, is harmful drinker and inactive but smokes'.
Execute.
fre DietDrinkPA.
*total n the same as total n for HealthScore.
```

## SmokePADiet

```
*Smoking, PA and Diet.
Do if SmokeScore=1 and Q1_1Less=1 and DietR=1 and ProbDrk=0.
Compute SmokePADiet=1.
Else if SmokeScore=1 and Q1_1Less=1 and DietR=1 and ProbDrk=1.
Compute SmokePADiet=0.
Else if SmokeScore=0 and (Q1_1Less=0 or Q1_1Less=1) and (DietR=0 or DietR=1) and
(ProbDrk=0 or ProbDrk=1).
Compute SmokePADiet=0.
Else if (SmokeScore=0 or SmokeScore=1) and Q1_1Less=0 and (DietR=0 or DietR=1) and
(ProbDrk=0 or ProbDrk=1).
Compute SmokePADiet=0.
Else if (SmokeScore=0 or SmokeScore=1) and (Q1_1Less=0 or Q1_1Less=1) and DietR=0 and
(ProbDrk=0 or ProbDrk=1).
Compute SmokePADiet=0.
End if.
Value labels SmokePADiet 0 'No combination' 1'Smokes,inactive and low fruit'.
Variable labels SmokePADiet 'Recruit smokes, is inactive and low fruit but is active'.
Execute.
fre SmokePADiet.
*total n the same as total n for HealthScore.
```

## RiskAll

```
Do if SmokeScore=1 and Q1_1Less=1 and DietR=1 and ProbDrk=1.
Compute RiskAll=1.
Else if SmokeScore=0 and (Q1_1Less=0 or Q1_1Less=1) and (DietR=0 or DietR=1) and
(ProbDrk=0 or ProbDrk=1).
Compute RiskAll=0.
```

```
Else if (SmokeScore=0 or SmokeScore=1) and Q1_1Less=0 and (DietR=0 or DietR=1) and
(ProbDrk=0 or ProbDrk=1).
Compute RiskAll=0.
Else if (SmokeScore=0 or SmokeScore=1) and (Q1_1Less=0 or Q1_1Less=1) and DietR=0 and
(ProbDrk=0 or ProbDrk=1).
Compute RiskAll=0.
Else if (SmokeScore=0 or SmokeScore=1) and (Q1_1Less=0 or Q1_1Less=1) and (DietR=0 or
DietR=1) and ProbDrk=0.
Compute RiskAll=0.
End if.
Value labels RiskAll 0 'No combination' 1'All 4 risk factors'.
Variable labels RiskAll 'Whether recruit has all 4 risk factors'.
Execute.
fre RiskAll.
fre HealthScore.
*Same % of recruits with all 4 risk factors.
```

## RiskNone

```
Do if SmokeScore=0 and Q1_1Less=0 and DietR=0 and ProbDrk=0.
Compute RiskNone=1.
Else if SmokeScore=1 and (Q1_1Less=0 or Q1_1Less=1) and (DietR=0 or DietR=1) and
(ProbDrk=0 or ProbDrk=1).
Compute RiskNone=0.
Else if (SmokeScore=0 or SmokeScore=1) and Q1_1Less=1 and (DietR=0 or DietR=1) and
(ProbDrk=0 or ProbDrk=1).
Compute RiskNone=0.
Else if (SmokeScore=0 or SmokeScore=1) and (Q1_1Less=0 or Q1_1Less=1) and DietR=1 and
(ProbDrk=0 or ProbDrk=1).
Compute RiskNone=0.
Else if (SmokeScore=0 or SmokeScore=1) and (Q1_1Less=0 or Q1_1Less=1) and (DietR=0 or
DietR=1) and ProbDrk=1.
Compute RiskNone=0.
End if.
Value labels RiskNone 0 'No combination' 1' No risk factors'.
Variable labels RiskNone 'Whether recruit has no risk factors'.
Execute.
fre RiskNone.

fre HealthScore.
*Same % of recruits with all 4 risk factors.
```

## HealthScoreR

```
Recode HealthScore (0=0)(1=0)(2=1)(3=2)(4=2) into HealthScoreR.
Value labels HealthScoreR 0'0 or 1 risk factors' 1'2 risk factors' 2'3 or 4 risk factors'.
Variable labels HealthScoreR 'Health score collapsed into fewer categories'.
Execute.
```

## AUDIT_grouped

```
Do if AUDIT_C=0.
Compute AUDIT_grouped=0.
Else if range(AUDIT_C,1,4).
Compute AUDIT_grouped=1.
Else if range(AUDIT_C,5,8).
Compute AUDIT_grouped=2.
Else if range(AUDIT_C,9,12).
Compute AUDIT_grouped=3.
End if.
Variable labels AUDIT_grouped ‘AUDIT-C score categorised’
Value labels AUDIT_grouped 0'Non-drinker' 1'Score of 1 to 4' 2'Score of5 to 8' 3'Score of 9 to
12'.
```

```
Execute.
```

# 7. Training injuries

## FirstInjury

```
Do if AUDIT_C=0.
*Recode 1st injury date variable to enable injury incidence to be calculated
(cases with text indicate no injury or missing data).
autorecode Date1st_present /into FirstDateRR.
execute.

Do If FirstDateRR = 398.
Compute FirstInjury=0.
Else If FirstDateRR = 396.
Compute FirstInjury=-9.
Else If FirstDateRR = 397 or FirstDateRR = 1.
Compute FirstInjury=-8.
Else If range(FirstDateRR,2,395).
Compute FirstInjury=1.
End if.
Missing values FirstInjury (-8,-9).
Value labels FirstInjury 0 'No injury' 1 'Injured'.
Variable labels FirstInjury ‘Whether recruit suffered an injury’.
Execute.

*Check frequencies of first injury*.
FREQUENCIES VARIABLES=FirstInjury
  /ORDER=ANALYSIS.
*Frequencies matched those in original dataset* - i.e. Excel file
```

## FirstInjuryR

```
Do if FirstInjury=1 and inj1_trainR=1.
Compute FirstInjuryR=1.
ELSE IF FirstInjury=1 and inj1_trainR=0.
Compute FirstInjuryR=0.
Else if Firstinjury=0.
Compute FirstInjuryR=0.
Else if FirstInjury<0.
Compute FirstInjuryR=-9.
End if.
Variable labels FirstInjuryR 'Whether first injury was training injury'.
Value labels FirstInjuryR 0'No training injury' 1'Training injury'.
Execute.
```

## Inj1_TimeOut_Train

```
Do if Inj1_TrainR=1.
Compute Inj1_TimeOut_Train = ctime.days(Inj1_OutcomeDate-Inj1_Date).
Else if Inj1_TrainR=0.
Compute Inj1_TimeOut_Train =0.
Variable labels Inj1_TimeOut_Train ‘Time lost due to 1st injury’.
End if.
```

## TimeLostTrain

```
*recode system missing values on TimeOut variable to enable total time out to be calculated.
RECODE Inj1_TimeOut_Train (SYSMIS=0).
RECODE Inj2_TimeOut_Train (SYSMIS=0).
RECODE Inj3_TimeOut_Train (SYSMIS=0).
```

```
RECODE Inj4_TimeOut_Train (SYSMIS=0).
RECODE Inj5_TimeOut_Train (SYSMIS=0).
RECODE Inj6_TimeOut_Train (SYSMIS=0).
RECODE Inj7_TimeOut_Train (SYSMIS=0).
RECODE Inj8_TimeOut_Train (SYSMIS=0).
RECODE Inj9_TimeOut_Train (SYSMIS=0).
Execute.

Compute TimeLostTrain = Inj1_TimeOut_Train + Inj2_TimeOut_Train + Inj3_TimeOut_Train +
Inj4_TimeOut_Train + Inj5_TimeOut_Train + Inj6_TimeOut_Train +
Inj7_TimeOut_Train + Inj8_TimeOut_Train + Inj9_TimeOut_Train.
Variable labels TimeLostTrain ‘Time lost from training injury’.
Execute.
*Cross checked with original dataset and they match*.
```

## ActiveTimeTrain

```
Compute ActiveTimeTrain = TrainingTime-TimeLostTrain.
Variable Labels ActiveTimeTrain 'Total active time spent in training'.
Execute.

DESCRIPTIVES VARIABLES=ActiveTimeTrain
  /STATISTICS=MEAN STDDEV MIN MAX.
*There are a number of negative values which appear to be real.
```

## RiskTimeTrain

```
*Convert negative values to MISSING.
Do if ActiveTimeTrain<0.
Compute RiskTimeTrain=-8.
Else if ActiveTimeTrain>=0.
Compute RiskTimeTrain=ActiveTimeTrain.
End if.
Variable Labels RiskTimeTrain 'Total active time spent in training - negative values removed'.
Missing values RiskTimeTrain (-8).
Execute.
```

## FirstInjuryExTrain

```
*Recode each InjuryDate variable to enable injuries to be totalled.
autorecode Inj1_Date /into FirstDateRRTrain.
execute.
Do if FirstDateRRTrain>0 and inj1_trainR=1.
Compute FirstInjuryExTrain=1.
Else if FirstDateRRTrain>0 and inj1_trainR=0.
Compute FirstInjuryExTrain=0.
End if.
Recode FirstInjuryExTrain (sysmis=0).
Execute.
Variable labels FirstInjuryExTrain 'Whether or not recruit suffered 2nd injury'.
Value labels FirstInjuryExTrain 0'No or missing' 1'Training injury' .
*matches original excel.
```

## TotalInjuriesTrain

```
*Calculate total number of injuries.
Compute
TotalInjuriesTrain=FirstInjuryExTrain+SecondInjuryExTrain+ThirdInjuryExTrain+FourthInjuryExT
rain+FifthInjuryExTrain+SixthInjuryExTrain
+SeventhInjuryExTrain+EighthInjuryExTrain+NinthInjuryExTrain.
Execute.

Temporary.
```

```
Select if FirstInjuryR>=0.
DESCRIPTIVES VARIABLES=TotalInjuriesTrain
  /STATISTICS=MEAN STDDEV SUM MIN MAX.

*CROSS-CHECK DERIVED TOTAL OF TOTAL INJURIES WITH ORIGINAL EXCEL DATA (by using counta function)
- both totals match (1785)
```

## IncTL_train

```
Do if FirstInjuryR=1 and Inj1_TimeOut>0.
Compute IncTL_train=1.
else if FirstInjuryR=1 and Inj1_TimeOut=0.
Compute IncTL_train=0.
else if FirstInjuryR=0.
Compute IncTL_train=-8.
else if FirstInjuryR=-8.
Compute IncTL_train=-8.
else if FirstInjuryR=-9.
Compute IncTL_train=-9.
End if.
Value labels IncTL_train 0'No' 1'Yes'.
Missing values IncTL_train (-8,-9).
Variable labels IncTL_train 'Whether 1st injury was a time-loss injury'.
Execute.
```

## TotalTimeLossTrain

```
Count TotalTimeLossTrain =  IncTL_train to IncTLninth_train(1 thru 10).
Execute.
```

## TimeInjAnyTrain

```
Do if FirstInjuryR=1.
Compute TimeInjAnyTrain=ctime.days(Inj1_Date-InputDate).
Else if FirstInjuryR=0 and DischargedR=1.
Compute TimeInjAnyTrain=ctime.days(CompleteDate-InputDate).
Else if FirstInjuryR=0 and DischargedR=2.
Compute TimeInjAnyTrain=ctime.days(DischargeDate-InputDate).
Else if FirstInjuryR<0.
Compute TimeInjAnyTrain=-9.
End if.
Variable labels TimeInjAnyTrain 'Time to first training injury'.
Missing values TimeInjAnyTrain (lo thru -1).
Execute.
```

## TimeInjTLTrain

```
Do if FirstInjuryR=1 and Inj1_TimeOut>0.
Compute TimeInjTLTrain=ctime.days(Inj1_Date-InputDate).
Else if FirstInjuryR=1 and Inj1_TimeOut=0 and Inj2_TimeOut>0.
Compute TimeInjTLTrain=ctime.days(Inj2_Date-InputDate).
Else if FirstInjuryR=1 and Inj1_TimeOut=0 and Inj2_TimeOut=0 and Inj3_TimeOut>0.
Compute TimeInjTLTrain=ctime.days(Inj3_Date-InputDate).
Else if FirstInjuryR=1 and Inj1_TimeOut=0 and Inj2_TimeOut=0 and Inj3_TimeOut=0 and
Inj4_TimeOut>0.
Compute TimeInjTLTrain=ctime.days(Inj4_Date-InputDate).
Else if FirstInjuryR=1 and Inj1_TimeOut=0 and Inj2_TimeOut=0 and Inj3_TimeOut=0 and
Inj4_TimeOut=0
and Inj5_TimeOut>0.
Compute TimeInjTLTrain=ctime.days(Inj5_Date-InputDate).
Else if FirstInjuryR=1 and Inj1_TimeOut=0 and Inj2_TimeOut=0 and Inj3_TimeOut=0 and
Inj4_TimeOut=0
```

```
and Inj5_TimeOut=0 and Inj6_TimeOut>0.
Compute TimeInjTLTrain=ctime.days(Inj6_Date-InputDate).
Else if FirstInjuryR=1 and Inj1_TimeOut=0 and Inj2_TimeOut=0 and Inj3_TimeOut=0 and
Inj4_TimeOut=0
and Inj5_TimeOut=0 and Inj6_TimeOut=0 and Inj7_TimeOut>0.
Compute TimeInjTLTrain=ctime.days(Inj7_Date-InputDate).
Else if FirstInjuryR=1 and Inj1_TimeOut=0 and Inj2_TimeOut=0 and Inj3_TimeOut=0 and
Inj4_TimeOut=0
and Inj5_TimeOut=0 and Inj6_TimeOut=0 and Inj7_TimeOut=0 and Inj8_TimeOut>0.
Compute TimeInjTLTrain=ctime.days(Inj8_Date-InputDate).
Else if FirstInjuryR=1 and Inj1_TimeOut=0 and Inj2_TimeOut=0 and Inj3_TimeOut=0 and
Inj4_TimeOut=0
and Inj5_TimeOut=0 and Inj6_TimeOut=0 and Inj7_TimeOut=0 and Inj8_TimeOut=0 and
Inj9_TimeOut>0.
Compute TimeInjTLTrain=ctime.days(Inj9_Date-InputDate).
Else if FirstInjuryR=1 and Inj1_TimeOut=0 and Inj2_TimeOut=0 and Inj3_TimeOut=0 and
Inj4_TimeOut=0
and Inj5_TimeOut=0 and Inj6_TimeOut=0 and Inj7_TimeOut=0 and Inj8_TimeOut=0 and
Inj9_TimeOut=0
and dischargedR=1.
Compute TimeInjTLTrain=ctime.days(CompleteDate-InputDate).
Else if FirstInjuryR=1 and Inj1_TimeOut=0 and Inj2_TimeOut=0 and Inj3_TimeOut=0 and
Inj4_TimeOut=0
and Inj5_TimeOut=0 and Inj6_TimeOut=0 and Inj7_TimeOut=0 and Inj8_TimeOut=0 and
Inj9_TimeOut=0
and dischargedR=2.
Compute TimeInjTLTrain=ctime.days(DischargeDate-InputDate).
Else if FirstInjuryR=0 and DischargedR=1.
Compute TimeInjTLTrain=ctime.days(CompleteDate-InputDate).
Else if FirstInjuryR=0 and DischargedR=2.
Compute TimeInjTLTrain=ctime.days(DischargeDate-InputDate).
Else if FirstInjuryR<0 .
Compute TimeInjTLTrain=-9.
End if.
Variable labels TimeInjTLTrain 'Time to first time loss injury'.
Missing values TimeInjTLTrain (lo thru -1).
Execute.
```

## FirstInjuryTrainR

```
autorecode Inj1_Date /into Inj_1DateTrain.
execute.
Do if Inj_1DateTrain>0 and inj1_trainR=1.
Compute FirstInjuryTrainR=1.
Else if Inj_1DateTrain>0 and inj1_trainR=0.
Compute FirstInjuryTrainR=2.
End if.
Recode FirstInjuryTrainR (sysmis=0).
Execute.
Variable labels FirstInjuryTrainR 'Whether or not recruit suffered 1st injury'.
Value labels FirstInjuryTrainR 0'No or missing' 1'Training injury' 2'Non-training injury'.
fre FirstInjuryTrainR.
*matches original excel.
```

## FirstInjuryTrainR

```
*new variable for time-loss injuries (denominator is all recruits, not all injuries).
Do if FirstInjuryR=1 and Inj1_TimeOut>0.
Compute IncTL_trainR=1.
else if FirstInjuryR=1 and Inj1_TimeOut=0.
Compute IncTL_trainR=0.
else if FirstInjuryR=0.
Compute IncTL_trainR=0.
```

```
else if FirstInjuryR<0.
Compute IncTL_trainR=-8.
End if.
Value labels IncTL_trainR 0'No' 1'Yes'.
Missing values IncTL_trainR (-8).
Variable labels IncTL_trainR 'Whether recruit suffered time-loss injury'.
Execute.
```

## InjWeek

```
Do if range(TimeInjAnyTrain,1,42).
Compute InjWeek=1.
Else if range(TimeInjAnyTrain,43,56).
Compute InjWeek=2.
Else if range(TimeInjAnyTrain,57,70).
Compute InjWeek=3.
Else if range(TimeInjAnyTrain,71,84).
Compute InjWeek=4.
Else if range(TimeInjAnyTrain,85,98).
Compute InjWeek=5.
Else if TimeInjAnyTrain>98.
Compute InjWeek=6.
Else if TimeInjAnyTrain=0.
Compute InjWeek=0.
End if.

Variable labels InjWeek 'Week in which first injury occurred'.
Value labels InjWeek 1'0-6' 2'6-8' 3'8-10' 4'10-12' 5'12-14' 6 '14-16'.
```